Representation based regression for object distance estimation

Mete Ahishali [a,*], Mehmet Yamac [a], Serkan Kiranyaz [b], Moncef Gabbouj [a]

[a] Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, 33720, Finland
[b] Department of Electrical Engineering, Qatar University, Doha, 2713, Qatar

ARTICLE INFO

ABSTRACT

In this study, we propose a novel approach to predict the distances of the detected objects in an observed scene. The proposed approach modifies the recently proposed Convolutional Support Estimator Networks (CSENs). CSENs are designed to compute a direct mapping for the Support Estimation (SE) task in a representation-based classification problem. We further propose and demonstrate that representation-based methods (sparse or collaborative representation) can be used in well-designed regression problems especially over scarce data. To the best of our knowledge, this is the first representation-based method proposed for performing a *regression* task by utilizing the modified CSENs; and hence, we name this novel approach as *Representation-based Regression (RbR)*. The initial version of CSENs has a proxy mapping stage (i.e., a coarse estimation for the support set) that is required for the input. In this study, we improve the CSEN model by proposing Compressive Learning CSEN (CL-CSEN) that has the ability to jointly optimize the so-called proxy mapping stage along with convolutional layers. The experimental evaluations using the KITTI 3D Object Detection distance estimation dataset show that the proposed method can achieve a significantly improved distance estimation performance over all competing methods. Finally, the software implementations of the methods are publicly shared at https://github.com/meteahishali/CSENDistance.

1. Introduction

Distance estimation has been a crucial task since its application plays a vital role in many autonomous frameworks, e.g., autonomous driving, unmanned aerial vehicles, and robotics. One can estimate the object specific distance from the depth scene produced by depth sensors such as LiDAR or utilizing such methods that use only visual information. Naturally, the latter is preferable because of the extra cost of LiDAR. Moreover, even though the LiDAR sensor can operate under varying weather conditions, it has a limited coverage area such as 5% of the image space (Wang et al., 2019). Hence, there have been various methods (Casser, Pirk, Mahjourian, & Angelova, 2019; Chang & Chen, 2018; Gökçe, Üçoluk, Şahin, & Kalkan, 2015; Haseeb, Guan, Ristić-Durrant, & Gräser, 2018; Mahjourian, Wicke, & Angelova, 2018; Wang et al., 2019; Zhu & Fang, 2019) that focus on developing computer vision solutions for the depth estimation including supervised and unsupervised approaches. For example, Chang and Chen (2018) has utilized multiple cameras to compensate for the lack of sensors. On the other hand, the need for multiple cameras and processing costs are disadvantages of a stereo-camera based depth estimation method. Thus, several methods

have studied monocular depth estimation (Casser et al., 2019; Mahjourian et al., 2018), and they have revealed that by following recent trends in neural networks, i.e., fully convolutional neural networks, depth estimation performance with a single RGB image can be comparable enough with a stereo-camera based approaches. As unsupervised learning strategies, Casser et al. (2019) and Mahjourian et al. (2018) propose to learn depth information from structural changes within consequent frames. Additionally, besides using the visual data alone, a hybrid approach combining and utilizing both visual and sensor data can be another alternative for enhancing the noisy or erroneous depth predictions. For example, the authors claim in Wang et al. (2019) that their method can be integrated into various learning-based methods that use visual information, and it can improve the performance of the methods by sparse LiDAR measurements.

Nevertheless, the aforementioned methods except (Gökçe et al., 2015; Haseeb et al., 2018; Zhu & Fang, 2019) have focused on producing dense depth maps which means computing a heatmap that gives a sense of relative depth distance information in an observed scene. On the other hand, the necessity of dense depth maps varies among applications, i.e., in an autonomous driving application, the distance information of the objects is more desirable than providing the depth map of the scene. There are only a few studies (Gökçe et al., 2015; Haseeb et al., 2018; Zhu & Fang, 2019) proposing object distance estimation for the objects in an observed scene.

The pioneer study (Haseeb et al., 2018) of object distance estimation on land proposes a two-tiers methodology: (i) first, detection of the location, and then the classification of an object. (ii) extraction of the related features such as the bounding box information (the width, the height, etc.), and class-specific ones (such as predefined average length of detected class). (iii) Finally, using a Multi-Layer Perceptron (MLP) to predict the camera distance of the bounding box in meters. However, their approach directly depends on the performance of object classification, while a misclassification of the given ROI may lead to a complete failure in distance estimation. Similarly, Gökçe et al. (2015) proposes to use only the geometric information of the bounding box as features and train a Support Vector Regressor (SVR). Next, in Zhu and Fang (2019), a Convolutional Neural Network (CNN) is used to extract representative features and these features have then been used for the regression and the classification tasks by two MLPs to predict the distance of the object and its category. Since the overall framework in Zhu and Fang (2019) is trained jointly by combining the classification and regression losses, the categorical information of the objects has boosted the estimation of the distance.

Overall, comparing with the recent improvements in dense depth estimation (Casser et al., 2019; Mahjourian et al., 2018; Wang et al., 2019), there is a lack of existing research focusing on object distance estimation. In this study, we believe that the importance of object distance estimation is obvious as the number of recent advances in the *state-of-the-art* object detectors has been growing and further analysis over these objects can provide better assistance to autonomous systems.

Deep Learning approaches with the recent advances in Convolutional Neural Networks (CNNs) have provided *state-of-the-art* performance levels in various computer vision tasks such as object detection, image recognition, and image segmentation. To achieve such performance levels, the deep learning-based approaches require a massive training dataset. On the other hand, the proposed solution for the object distance estimation task should be suitable to work with relatively small-scale annotated data. For example, one can compare KITTI 3D Object Detection (Geiger, Lenz, & Urtasun, 2012) dataset having annotated 7481 scenes with Imagenet (Russakovsky et al., 2015) having over a million samples.

To address this need, in this study, we first formulate the distance estimation problem as a multi-class classification task by quantizing the distance in meters and use representation-based classification techniques including two categories: Sparse Representation-based Classification (SRC) and Collaborative Representation-based Classification (CRC). The approaches for SRC (Wright et al., 2010; Wright, Yang, Ganesh, Sastry, & Ma, 2008) and CRC (Zhang, Yang, & Feng, 2011) are well suited for the limited data and they are commonly used for the classification in the existing studies as follows. A representative dictionary $\mathbf{D}$ is constructed by grouping training samples column-wise. In the inference phase, a test sample $\mathbf{y}$ will be attempted to be represented by the linear combination of the atoms of the formed dictionary $\mathbf{D}$, i.e., solving $\mathbf{y} = \mathbf{Dx}$ for $\mathbf{x}$ where $\mathbf{x}$ is a vector of representation coefficients. Accordingly, in SRC methods, it is aimed to find a sparse $\hat{\mathbf{x}}$ (just have enough non-zero components so that the query sample is represented with a small error margin). Alternatively, in CRC, the least-square sense solution is applied, i.e., $\hat{\mathbf{x}} = \left(\mathbf{D}^T \mathbf{D} + \lambda \mathbf{I}\right)^{-1} \mathbf{D}^T \mathbf{y}$, where λ is the regularization parameter. Overall, the same motivation is valid for both categories: the atoms having higher estimated representation coefficient values, $\hat{\mathbf{x}}$, are likely to have the same class label with the query sample $\mathbf{y}$. It has been observed in Zhang et al. (2011) that the CRC approach has provided marginally reduced classification performance compared to SRC methods. However, the computational complexity of

the methods that rely on SRC is significant considering that they require iterative computations to solve the problem.

In this study, we propose the following approach of using a representation-based scheme in the object distance estimation task. First, the cropped objects are resized to have fixed size images for each object, and then their corresponding features are obtained by using pre-trained networks DenseNet-121 (Huang, Liu, Van Der Maaten, & Weinberger, 2017), VGG19 (Simonyan & Zisserman, 2014), and ResNet-50 (He, Zhang, Ren, & Sun, 2016) over the ImageNet dataset. Next, a dictionary is created with atoms of relative features that are from the classes obtained by discretizing the distances of the object. Finally, a representation-based classification method is applied to detect the class which will correspond to the discretized distance of the query object. The main advantage of the proposed approach is that the categorical information of the object is not used in the distance estimation unlike the methods in Haseeb et al. (2018), Zhu and Fang (2019); hence the classification performance of a single-stage object detector does not affect the distance prediction performance.

As an alternative approach, we propose to consider this as a regression problem. In order to make a direct distance estimation without discretization during the inference phase, we modify Convolutional Support Estimator Network (CSEN) that was originally proposed as a representation-based classifier in Yamac, Ahishali, Kiranyaz, and Gabbouj (2020). The CSEN approach combines the conventional representation-based classification technique with the learning-based approach involving CNNs. We define the task of Support Estimation (SE) to estimate locations of the non-zero components of $\mathbf{x}$. Indeed, the support of the non-zero coefficient forms sufficient information to obtain the class of the query sample. The previous works (Ahishali et al., 2021; Yamaç et al., 2021; Yamac et al., 2020) have shown that CSENs provide *state-of-the-art* classification performance levels and their computational complexity are insignificant since they can directly map the support set of the query sample. Moreover, they are well-suited for limited annotated data since they do not have the tendency to *overfit* due to their compact structures. Up to date, the CSEN approach has never been designed and evaluated for a regression task. In this study, we show that using the modified CSEN configuration, it is possible to perform a regression task that is henceforth called as *Representation-based Regression (RbR)*. Finally, we propose further improvements over the CSEN framework. The initial CSEN version (Yamac et al., 2020) has required the so-called proxy, $\tilde{\mathbf{x}}$, estimation based on the least-square solution, i.e., $\tilde{\mathbf{x}} = \left(\mathbf{D}^T \mathbf{D} + \lambda \mathbf{I}\right)^{-1} \mathbf{D}^T \mathbf{y}$. In this study, we propose an end-to-end learning, the so-called Compressive Learning CSEN (CL-CSEN) framework that *jointly* optimizes the proxy mapping and SE estimation.

Overall, the novel and significant contributions of this study can be summarized as follows:

- Representation-based classification approaches are used in an object-specific distance estimation task for the first time.
- To the best of our knowledge, this is the first study that formulates a regression task in the form of representation-based estimation approach. The proposed methodology is henceforth named as *Representation-based Regression (RbR)*.
- With the proposed approach, the *state-of-the-art* performance level is achieved using compact configurations and a non-iterative SE. This does not only enables an accurate estimation with a limited number of annotated data, it further yields an elegant efficiency in terms of computational complexity.
- The improved framework with CL-CSEN enables the joint optimization of SE framework with the denoiser matrix $\mathbf{B}$.

Accordingly, the new CL-CSEN and initial CSEN models have different input and output pairs. The latter requires so-called proxy computation for the input, i.e., $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{y}$, whereas CL-CSEN is an end-to-end framework that can directly map the compressed query feature vector $\mathbf{y}$ to the estimated distance $\hat{d}$. In MLP part of the CL-CSEN model, we initialize the weights of the fully-connected layer with $\mathbf{B}^T$ with $\mathbf{B} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T$.

- Finally, the proposed approach performs object-specific distance estimation, whereas the majority of the aforementioned studies focus on dense depth map estimation producing depth masks for an observed scene. Only a limited number of studies has been proposed for object-specific distance estimation including (Gökçe et al., 2015; Haseeb et al., 2018; Zhu & Fang, 2019). Furthermore, the proposed method is designed to be run on a single RGB image, while the studies (Haseeb et al., 2018; Zhu & Fang, 2019) require additional information such as the object class information and camera projection matrix.

Our experimental evaluations over the KITTI benchmark dataset (Geiger et al., 2012) show that the distance estimation performance with the proposed CSEN approach outperforms all competing methods, i.e., the competing distance estimator SVR (Gökçe et al., 2015) and alternative representation-based approaches including CRC (Zhang et al., 2011) and SRC approaches (Wright et al., 2010, 2008). Moreover, although the direct comparison of the proposed approach is not fair against (Zhu & Fang, 2019) due to the reasoning mentioned earlier, the proposed approach still outperforms considering their reported performance metrics.

The rest of the paper is structured as the following: the theoretical background and the prior art will be presented in Section 2. Then, the proposed object distance estimation with CSEN and CL-CSEN will be detailed in Section 3. Next, the experimental evaluations over the KITTI dataset are presented in Section 4. Finally, concluding remarks will be drawn in Section 6.

2. Background and prior art

In this section, we shall first provide a brief background of sparse representation, then, discuss the representation-based classification theory including SRC and CRC methods.

The following notations and terms are defined in this study. For a vector, $\mathbf{x} \in \mathbb{R}^n$, the ℓ_p-norm is $\|\mathbf{x}\|_{\ell_p^n} = \left(\sum_{i=1}^{n} |x_i|^p\right)^{1/p}$ where $p \geq 1$, whereas the ℓ_0-norm and ℓ_∞-norm are defined as $\|\mathbf{x}\|_{\ell_0^n} = \lim_{p\to 0} \sum_{i=1}^{n} |x_i|^p = \#\{j : x_j \neq 0\}$ and $\|\mathbf{x}\|_{\ell_\infty^n} = \max_{i=1,\ldots,n}(|x_i|)$ for the vector $\mathbf{x}$, respectively. Let a signal $\mathbf{s}$ is sparsely represented in a domain Φ such that $\mathbf{s} = \Phi\mathbf{x}$ where $\|\mathbf{x}\|_0 \leq k$, then it is said that the signal $\mathbf{s}$ is strictly k-sparse since it can be represented using less than $k+1$ non-zero coefficients in a proper domain. That is to say, it is possible to represent the signal $\mathbf{s}$ with only a few basis vectors in a proper domain Φ. The sparse support set Λ is then a set that contains locations of these non-zero coefficients of $\mathbf{x}$ such that $\Lambda := \{i : x_i \neq 0\}$ and $\Lambda \subset \{1, 2, 3, \ldots, n\}$.

Let $\mathbf{A}$ is a subspace for the signal $\mathbf{s}$ such that $\mathbf{y} = \mathbf{A}\mathbf{s}$. Accordingly, a signal $\mathbf{y}$ can be projected to the subspace $\mathbf{A}$ as follows:

$$\mathbf{y} = \mathbf{A}\mathbf{s} = \mathbf{A}\Phi\mathbf{x} = \mathbf{D}\mathbf{x}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{m \times d}$ is called compression matrix, $\mathbf{D} \in \mathbb{R}^{m \times n}$ is the equivalent dictionary, and $m << n$; and hence the corresponding system is underdetermined. We necessitate a priori information regarding the unknown $\mathbf{x}$ to solve such an ill-posed problem in (1) since it is non-uniquely solvable. Donoho and Elad (2003) has shown that at least k-sparse signal pairs in a sparsifying basis Φ are distinguishable in the dictionary $\mathbf{D}$ if $\mathbf{D}$ satisfies some

properties. Consequently, it immediately indicates that the below solution is unique,

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ subject to } \mathbf{D}\mathbf{x} = \mathbf{y}, \tag{2}$$

if $\|\mathbf{x}\|_0 \leq k$, $m \geq 2k$, and the minimum number of linearly independent columns in $\mathbf{D}$ is also greater than $2k$ (Donoho & Elad, 2003).

On the other hand, the solution of (2) is NP hard and the problem is non-convex. Fortunately, we can relax the optimization problem, ℓ_0-minimization, to its closest norm based problem defined as Basis Pursuit (Chen, Donoho, & Saunders, 2001) that is ℓ_1-norm:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \ s.t. \ \mathbf{x} \in \mathcal{U}(\mathbf{y}) \tag{3}$$

where $\mathcal{U}(\mathbf{y}) = \{\mathbf{x} : \mathbf{D}\mathbf{x} = \mathbf{y}\}$. The equivalent to the one of the sparse representation problem (2), but more tractable solution can be achieved by solving ℓ_1-minimization defined in (3) under some conditions such as $m > k(\log(n/k))$ and $\mathbf{D}$ satisfied Restricted Isometry Property (Candes, 2008).

2.1. Generic sparse support estimation (SE)

Support estimation can be defined as finding the non-zero locations of a corresponding sparse signal. Indeed, in many practical problems, the full signal recovery; the recovery of the signal magnitude, sign, and support set, may not be necessary. For example, in a representation-based classification problem (Wright et al., 2010, 2008; Zhang et al., 2011), estimating the locations of non-zero elements in $\mathbf{x}$ so-called the support set, Λ, is enough to determine the corresponding class. For a linear feed-forward model, $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{z}$, with an additive noise $\mathbf{z}$, the works in the literature targeting SE from y are based on by first applying a signal recovery method then applying component-wise thresholding over the estimated signal, $\hat{\mathbf{x}}$, to compute $\hat{\Lambda}$. Accordingly, they can be divided into three categories depending on their reconstruction schemes: (i) estimators that are based on ℓ_1-minimization, (ii) least-square sense approximate methods such as LMSEE (Reeves & Gastpar, 2012), $\hat{\mathbf{x}}^{\text{LMMSE}} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}_{n\times n}\right)^{-1}\mathbf{D}^T\mathbf{y}$ and Maximum Correlation (MC) (Fletcher, Rangan, & Goyal, 2009), $\hat{\mathbf{x}}^{\text{MC}} = \mathbf{D}^T\mathbf{y}$, and (iii) Deep Neural Networks.

The approaches in (i) work in an iterative manner and they are computationally costly; and hence, not efficient if the aim is to only recover support information. The methods in (ii) are non-iterative and direct approaches but their performances are limited compared to the previous ones (see Reeves & Gastpar, 2012 for a detailed discussion). Finally, deep learning-based approaches (Borgerding, Schniter, & Rangan, 2017) in the group (iii) target a direct mapping for the signal reconstruction task. However, the major concern is that the signal reconstruction task is harder than SE, and it requires deep networks having complex architectures with millions of parameters to enable a direct mapping. This further requires a massive size of training data for a proper generalization. Furthermore, these deep unfolding networks (Borgerding et al., 2017) consist of dense layers making them computationally intensive and more sensitive to the additional noises (Yamac et al., 2020). As a remedy, our recent approach, CSEN (Yamac et al., 2020), which can perform direct SE without first applying signal recovery, provides an alternative and computationally efficient solution. The compact design of CSEN enables elegant performance even with small-scale training data. Moreover, compared to deep networks with dense layers, CSEN with convolution layers provide robust SE in noisy cases. For a more detailed analysis, the readers are referred to Yamac et al. (2020) where we compare the performances of the traditional support estimators with the proposed CSEN approach and we address the major limitations and drawbacks with the classical support estimator methods.

2.2. Representation-based classification

As discussed earlier, in representation-based classification task, predicting the locations of the non-zero elements in **x** is more important than computing the exact values. In the following sub-sections, we shall provide a brief explanation about how SRC and CRC methods perform classification. Basically, SRC methods are in the aforementioned first group of support estimators, whereas the CRC method belongs to the second group.

2.2.1. Sparse representation-based classification

When a test sample **y** is introduced, the query sample **y** is tried to be represented as a linear combination of the atoms of the dictionary **D**. In general, SRC methods estimate sparse representation coefficients $\hat{\mathbf{x}}$ that only a few non-zero coefficients exist to represent the query sample. We expect that these active components of $\hat{\mathbf{x}}$ will correspond to the samples having the same label as the test sample. There are many existing studies that utilize the SRC approach for various classification tasks such as face recognition (Wright et al., 2010), coronavirus disease 2019 (COVID-19) recognition (Yamaç et al., 2021), early COVID-19 detection (Ahishali et al., 2021), human action recognition (Guha & Ward, 2011), and hyper-spectral image classification (Li & Du, 2016).

In the previous discussion regarding (2) and (3), the statements were valid for the exactly k-sparse signal pairs, whereas in practice, the signal **x** may not be exactly k-sparse due to the modeling errors or noise in the data. Consequently, given the measurement with the additive noise: $\mathbf{y} = \mathbf{Dx} + \mathbf{z}$, the exact recovery of the signal is unfeasible. However, the stable signal recovery is still possible in which the stable recovery refers that $\hat{\mathbf{x}}$ obeys $\|\mathbf{x} - \hat{\mathbf{x}}\| \leq \kappa \|\mathbf{z}\|$ hold for the estimated sparse signal $\hat{\mathbf{x}}$, where κ is a relatively small constant. For instance, it is provided in Candes and Plan (2011) that using the following so-called Lasso formulation:

$$\min_{\mathbf{x}} \left\{ \|\mathbf{Dx} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\} \tag{4}$$

the partial recovery of the sparse **x** is achievable. Correspondingly, it is also proven that ℓ_1 solution can still provide exact computing of **x** in noise-free cases.

Wright et al. (2010) proposes to follow a four-step approach instead of using (4) directly: (i) normalize all the atoms in **D** and **y** to have unit ℓ_2-norm, (ii) apply the signal reconstruction step: $\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1$ s.t. $\|\mathbf{y} - \mathbf{Dx}\|_2$, (iii) residual finding: $\mathbf{e}_i = \|\mathbf{y} - \mathbf{D}_i\hat{\mathbf{x}}_i\|_2$, where $\hat{\mathbf{x}}_i$ is the estimated coefficients corresponding the class i, (iv) estimated label: Class $(\mathbf{y}) = \arg\min (\mathbf{e}_i)$. Although this four-step solution introduces an additional residual finding step, it provides performance improvements over direct SE with (4) since the samples from different classes are actually correlated as in real life. The other SRC techniques (Guha & Ward, 2011; Li & Du, 2016) have followed similar multi-step approaches for the class estimation.

2.2.2. Collaborative representation-based classification

The study in Zhang et al. (2011) proposes to follow ℓ_2-minimization instead of ℓ_1-minimization in (4) as follows:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \|\mathbf{y} - \mathbf{Dx}\|_2^2 + \lambda \|\mathbf{x}\|_2^2 \right\} \tag{5}$$

Hence, they form the CRC approach by changing the second step of the four-step solution in Wright et al. (2010) with the following closed-form solution: $\hat{\mathbf{x}} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}_{n \times n} \right)^{-1} \mathbf{D}^T\mathbf{y}$. The motivation is that for a given a query signal **y** or vectorized image, the computed $\hat{\mathbf{x}}$ should have minimum energy with relatively small coefficients that correspond to samples in the dictionary **D** from the same class with the query **y**. Hence, due to the least-square

sense minimization technique, a collaborative representation is sought between the atoms of the dictionary. In representation-based classification scheme, the dictionary **D** mostly fails to satisfy the defined exact or robust recovery properties due to the correlation between samples. It is indeed discussed in Zhang et al. (2011) that if formulating the problem with the collaborative representation operates the classification rather than the sparse representation.

It is reported that the followed ℓ_2-minimization based solution provides especially high classification performances for a high compression ratio that is defined as m/d. In those cases, the CRC approach can even produce comparable or better classification results comparing with SRC. Note the fact that the CRC approach is considerably faster due to the presence of the closed-form solution in (5).

3. The proposed methodology

In the sequel, we will introduce the feature extraction procedure and the framework about using classical representation-based classification methods: SRC and CRC on the distance estimation task with the quantization. Then, the proposed CSEN based regression approach will be presented to directly predict the distance information without the quantization in the inference. Finally, a novel CL-CSEN framework will be introduced that is specifically designed to jointly optimize the denoiser and the regression parts of the CSEN during the training phase.

3.1. Estimation via representation-based classification

The representative dictionary that is needed to form in the representation-based classification methods can be formed by vectorized samples. However, we have revealed in Yamac et al. (2020) that for some cases, the atoms of the collected dictionary for a representation-based classification method are not representative enough if they are formed by directly putting the vectorized raw images. Hence, in the regression task as well, we propose to use a pre-trained CNN to produce more representative information.

The selected pre-trained models for this feature extraction procedure are DenseNet-121 (Huang et al., 2017), VGG19 (Simonyan & Zisserman, 2014), and ResNet-50 (He et al., 2016) that are trained over the ImageNet dataset with more than one million images:

- DenseNet-121 is a fully connected convolutional network: an L-layer DenseNet-121 has a total of $L(L + 1)/2$ connections whereas the corresponding traditional version of the convolutional network would have only L connections.
- VGG19 is a deep neural network consisting of convolutional and fully connected (dense) layers (as a generic CNN structure) without any skip-connections.
- ResNet-50 is based on residual learning having skip-connections between every other layer in the network.

Overall, DenseNet-121 and ResNet-50 are in the form of convolutional layers consisting of only convolutional layers except for the output layer, whereas VGG19 has multiple fully connected layers.

Accordingly, we compose the features before the last convolutional layers of DenseNet-121 and ResNet-50 and before the fully connected layers of VGG19. Then, the collected multiple feature maps are flattened by applying global max-pooling operation. Consequently, the described feature extraction procedure provides the mapping $\phi : \mathbb{R}^{N \times N \times 3} \rightarrow \mathbb{R}^d$ to produce a feature vector, $\mathbf{s}_i = \phi(\mathbf{I}_i)$, where $\mathbf{I}_i$ is the ith object cropped from the observed frame and resized using bilinear interpolation to a predetermined size i.e., $N \times N$ as demonstrated in Fig. 1. The
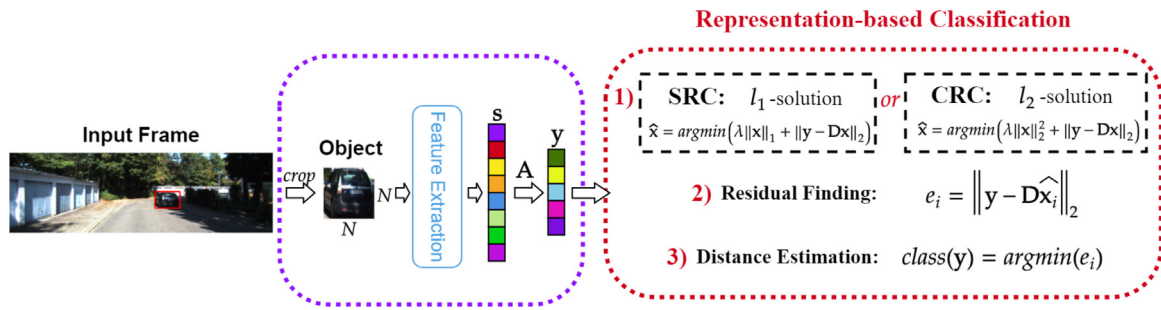
Fig. 1. The proposed framework for the object distance estimation is based on representation-based classification methodologies including Sparse Representation-based Classification (SRC) and Collaborative Representation-based Classification (CRC). The output class estimation yields the quantized estimated distance.

feature vector dimension $d = 1024$, 512, and 2048 for DenseNet-121, VGG19, and ResNet-50, respectively. The composed features for n number of objects are collected column-wise to have the matrix, $\boldsymbol{\Phi} \in \mathbb{R}^{d \times n}$. The representative dictionary **D** is then formed as $\mathbf{D} = \mathbf{A}\boldsymbol{\Phi}$ using the compression matrix $\mathbf{A} \in \mathbb{R}^{m \times d}$ as PCA. When forming the dictionary, we quantize the distances to interpret the regression problem as a classification problem. For example, let the desired sensitivity is selected as 1 m, then there would be 60-classes for a distance estimation task for the range of 1–60 m. This dictionary formation procedure is illustrated in Fig. 2. Apparently, the distance information in the extracted features comes from the resolution of the cropped input images since the distant objects tend to have blurry appearances due to the rescaling small-scale distant objects as observed in Fig. 2. Next, representation-based classification approaches with SRC and CRC can be used to predict the class which will correspond to the quantized distance. As illustrated in Fig. 1, the aforementioned four-step approach in Section 2 is used in the framework including the residual finding step.

3.2. The proposed representation-based regression (RbR) with CSENs

With the proposed approach, it is possible to produce exact estimates instead of quantized distances during the inference. Hence, to the best of our knowledge, as the first time in the literature, we are introducing the utilization of a representative dictionary for a complete regression task. Accordingly, the proposed approach that will be detailed next is called *Representation-based Regression (RbR)*.

Since the traditional approaches first fully reconstruct the signal before the actual SE task, the performance of the SE becomes highly dependent on the performance of the signal recovery. As discussed earlier, the signal reconstruction is not guaranteed if the required sparsity of **x** does not hold due to practical reasons such as the presence of significant noise or high correlation between samples as observed in some classification problems such as face recognition. Nevertheless, it is still possible to recover Λ fully (Rad, 2011; Scarlett & Cevher, 2016; Wainwright, 2007; Wang, Wainwright, & Ramchandran, 2008) or partially (Reeves & Gastpar, 2008, 2013; Scarlett & Cevher, 2016). With this motivation, we aim to learn a direct mapping to the corresponding support set $\hat{\Lambda}$ for a given query sample **y**.

The proposed SE follows a compact architecture that also maximizes the performance with a minimum number of annotated data. To this end, compact CSENs used in the previous work (Yamac et al., 2020) have been modified to enable regression in the distance estimation task. The proposed modified CSEN inherits the same capabilities and advantages. On the other hand, the initial network was a fully convolutional network consisting of only convolutional layers. In this study, we keep this strategy for SE as well and use MLP only for the regression over the predicted
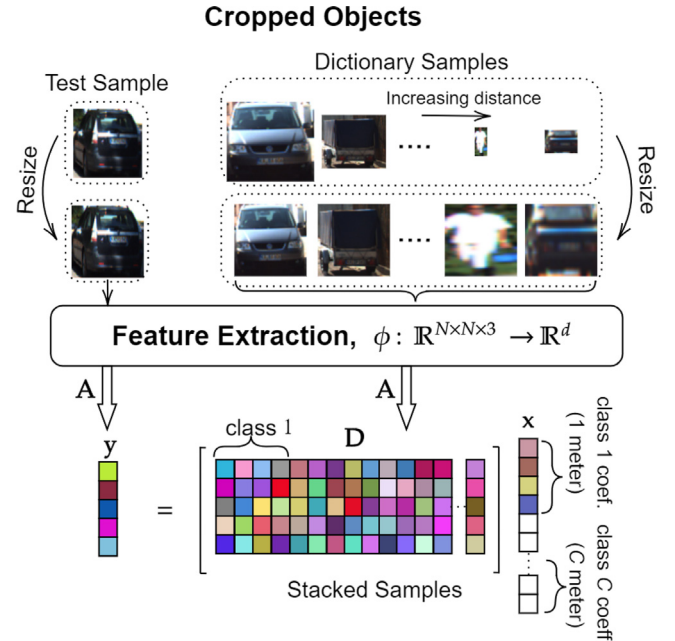

Fig. 2. To form the representative dictionary **D**, samples are collected with the increasing order of the distances. Then, they are resized and fed to the feature extractor. Next, after additional dimensional reduction operation with the matrix **A**, they are stacked in such a way that the first-class category corresponds to 1 m and the Cth class to C meters.

support sets. One may consider using an MLP-like structure for SE which is originally proposed for SR (Borgerding et al., 2017). However, the followed topology in CSENs brings several advantages over MLPs as proven by Yamac et al. (2020): low computational complexity, robustness to the noise, and learning capability with a limited amount of training data thanks to the compact structure of CSEN and significantly less number of parameters compared to the MLP.

A CSEN network is designed to produce a binary mask $\mathbf{v} \in \{0, 1\}^n$ by the following mapping $\mathcal{P}(\mathbf{y}, \mathbf{D}) : \mathbb{R}^n \mapsto [0, 1]^n$. Accordingly, it produces a probability vector **p** of each index to be counted as a support. The estimated support set, $\hat{\Lambda} = \{i \in \{1, 2, \ldots, n\} : \hat{v}_i = 1\}$, is then obtained by thresholding **p** with a fixed threshold. During the training phase, CSEN takes $\tilde{\mathbf{x}}$ as the input and produces $\hat{\mathbf{v}}$ as the SE, where $\hat{\mathbf{v}}, \tilde{\mathbf{x}} \in \mathbb{R}^n$; hence the learned transformation would be $\hat{\mathbf{v}} \leftarrow \mathcal{P}(\tilde{\mathbf{x}})$. Here, the input of CSEN is a rough estimation and it is called *proxy*. The proxy $\tilde{\mathbf{x}}$ can be the Maximum Correlation $\tilde{\mathbf{x}} = \mathbf{D}^T \mathbf{y}$ or LMMSE (Reeves & Gastpar, 2012) $(\mathbf{D}^T \mathbf{D} + \lambda \mathbf{I})^{-1} \mathbf{D}^T \mathbf{y}$. The input proxy $\tilde{\mathbf{x}}$ is then reshaped to a 2-D plane and convolved with the weight kernels $\{\mathbf{w}_1^1, \mathbf{w}_1^2, \ldots, \mathbf{w}_1^N\}$. After the addition of biases $\{b_1^1, b_1^2, \ldots, b_1^N\}$,
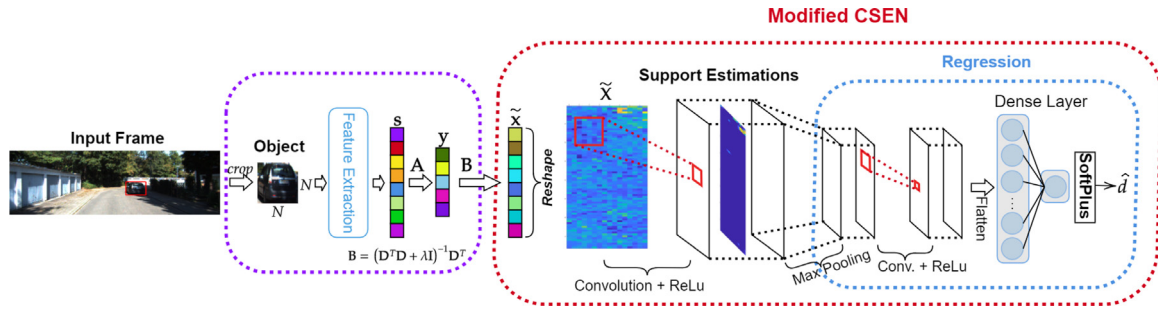
Fig. 3. The proposed framework for the object distance estimation based on Convolutional Support Estimator Networks (CSEN). The modified CSEN performs regression over the estimated support sets using the reshaped proxy signal $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{y}$ where $\mathbf{B} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T$.

the feature tensor $\mathbf{F}_1 = \{\mathbf{f}_1^1, \mathbf{f}_1^2, \ldots, \mathbf{f}_1^N\}$ in the first hidden layer with N number of weight kernels is formed:

$$\mathbf{F}_1 = \{S(\text{ReLu}(b_1^i + \mathbf{w}_1^i * \tilde{\mathbf{x}}))\}_{i=1}^N, \tag{6}$$

where $S(.)$ is the down- or up-sampling operation and $\text{ReLu}(x) = \max(0, x)$. This is illustrated in Fig. 3. At the layer l, the kth feature can be defined as follows:

$$\mathbf{f}_l^k = S(\text{ReLu}(b_l^k + \sum_{i=1}^{N_{l-1}} \mathbf{w}_l^{i,k} * \mathbf{f}_{l-1}^i)). \tag{7}$$

Accordingly, an L-layer CSEN network would have the following trainable weight and bias $\{\mathbf{w}, b\}$, parameters: $\Theta_{\text{CSEN}} = \left\{\{\mathbf{w}_1^i, b_1^i\}_{i=1}^{N_1}, \{\mathbf{w}_2^i, b_2^i\}_{i=1}^{N_2}, \ldots, \{\mathbf{w}_L^i, b_L^i\}_{i=1}^{N_L}\right\}$.

In SRC, the dictionary is collected by stacking training samples, for example, by concatenating the same class samples together. Thus, group ℓ_1-minimization can be used instead of (4):

$$\min_{\mathbf{x}} \left\{ \|\mathbf{D}\mathbf{x} - \mathbf{y}\|_2^2 + \lambda \sum_{i=1}^c \|\mathbf{x}_{G,i}\|_2 \right\} \tag{8}$$

where $\mathbf{x}_{G,i}$ is the group of coefficients from class i. Therefore, the cost function for a CSEN can be expressed as,

$$E(\mathbf{x}) = \sum_p (\mathcal{P}_\Theta(\tilde{\mathbf{x}})_p - v_p)^2 + \lambda \sum_{i=1}^c \left\| \mathcal{P}_\Theta(\tilde{\mathbf{x}})_{G,i} \right\|_2. \tag{9}$$

where $\mathcal{P}_\Theta(\tilde{\mathbf{x}})_p$ and v_p are the actual output and binary mask of the sparse code $\mathbf{x}$ for pth pixel, respectively.

The introduced regularization may bring additional computational complexity; hence, in the previous study (Yamac et al., 2020), an approximation of (9) is adopted for CSEN by applying average pooling over the output and then performing SoftMax operation to produce the class probabilities directly. However, for the regression problem, which is undertaken in this study, we propose to modify the architecture by replacing the average pooling with the max pooling and inserting an additional convolutional layer and fully connected layer right after the max-pooling as illustrated in Fig. 3. The included layers form the regression part of the modified CSEN. Then, the loss function of the modified CSEN for the regression can be expressed as $\mathcal{L}_{\text{CSEN}} = \sum_{i \in M} \text{smooth}_{\ell_1}(\mathcal{P}_\Theta(\tilde{\mathbf{x}}_i) - d_i)$ over a batch M, where $\mathcal{P}_\Theta(\tilde{\mathbf{x}}_i)$, d_i are the predicted and real distance values for the ith object and smooth ℓ_1-loss is expressed as,

$$\begin{cases} \text{smooth}_{\ell_1}(x) = 0.5x^2 & \text{if } |x| < 1 \quad \text{(a)} \\ |x| - 0.5 & \text{else .} \qquad\qquad\quad \text{(b)} \end{cases} \tag{10}$$

The selected loss function combines ℓ_1 and ℓ_2 penalizations such that if the absolute value of the error is smaller than 1, it behaves like ℓ_2-loss. In this way, oscillations are prevented when the model state is closer to the convergence point. On the other
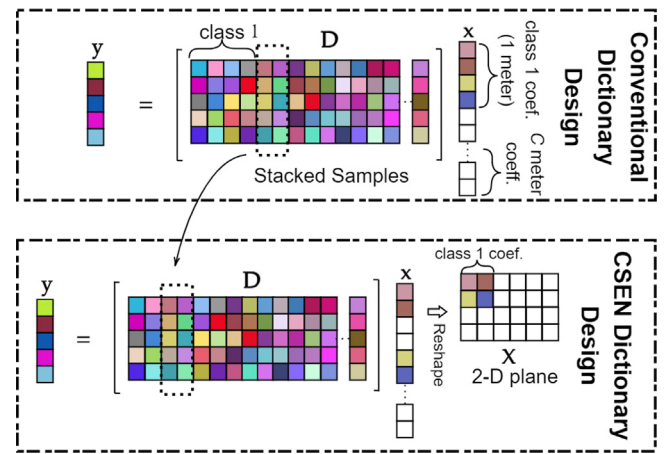

Fig. 4. Conventional dictionary design versus the proposed dictionary design for the CSEN. In the conventional dictionary design, samples are collected with the increasing order of the distances. The first, second, third class categories correspond to 1 m, 2 m, 3 m, respectively, and the Cth class corresponds to C meters.

hand, if the absolute value is large, thanks to the ℓ_1 behavior, the loss function is more robust to outliers and provides more stable gradients. The input proxy is selected as $\tilde{\mathbf{x}}_i = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T\mathbf{y}_i$ from LMMSE where $\mathbf{y}_i = \mathbf{A}\mathbf{f}_i$ is obtained for the extracted object feature $\mathbf{f}_i = \phi(\mathbf{I}_i)$, and the input and output pair of the proposed method for the regression is $(\tilde{\mathbf{x}}^{\text{train}}, d^{\text{train}})$ for the training.

Note the fact that the proposed RbR method can directly map the exact distance values and it is possible to train the model using the exact distance information. The quantized distances are only used when forming the dictionary $\mathbf{D}$ with the selected quantized dictionary samples. In Section 3.1, we have detailed the distance estimation utilizing representation-based classification with SRC and CRC approaches since they can only estimate the quantized distances that correspond to a classification task, i.e., class c corresponds to objects of c meter away from the camera. Therefore, the grouped features from different objects (e.g., car, person, and truck), but from the same distances (c-meter) as shown in Fig. 2. In this way, we will have a categorical invariant distance estimator unlike the literature work (Haseeb et al., 2018; Zhu & Fang, 2019).

In the traditional approaches with SRC and CRC, one can directly use the collected representative dictionary $\mathbf{D}$ having the samples collected in a random order as long as the ordering is known since the recovery of $\mathbf{x}$ is obtained from $\mathbf{y} = \mathbf{D}\mathbf{x}$. However, in the CSENs, direct mapping from $\mathbf{y}$ is performed using 2-D convolutional layers. Hence, in the proposed CSENs, it is important to group samples with the same quantized distances together after reshaping the proxy $\tilde{\mathbf{x}}$ since the grouped coefficients are max
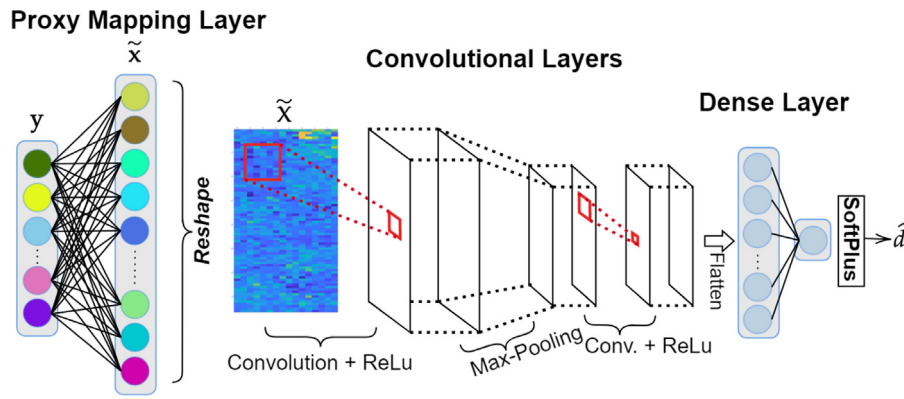
Fig. 5. The proposed Compressive Learning CSEN (CL-CSEN) framework that jointly optimizes proxy mapping with support estimation and regression parts during the training.

pooled in the feed-forward phase as discussed. Accordingly, the columns of the dictionary **D** are re-ordered in such a way that after reshaping the proxy into a 2-D plane, the samples with the same distances in the quantized level are grouped together. This proposed re-ordering topology is illustrated in Fig. 4 where 1-D coefficient vector **x** is reshaped to a 2-D plane that yields **X**. Correspondingly, one can directly say that the input size of the CSEN depends on the collected dictionary size and the stride size (also kernel size) of the average pooling depends on the number of samples within the same distance level.

3.3. Compressive learning CSEN (CL-CSEN) approach

In the CSEN approach, the input is the reshaped proxy signal, $\tilde{\mathbf{x}}$, which is obtained directly by $\tilde{\mathbf{x}} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T\mathbf{y}$. Ultimately, the performance of the CSEN was therefore limited to this proxy mapping stage i.e., $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{y}$ since **B** is treated as a constant during the training. To overcome this limitation, we propose to fine-tune the denoiser matrix **B** as follows: we include two additional fully-connected (dense) layers right before the first convolutional layer of the CSENs. The neurons connecting the input layer to the first hidden dense layer are initialized with $\mathbf{B}^T$ where $\mathbf{B} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T$. Next, the output of the first hidden dense layer is reshaped to form the input of the first hidden convolutional layer.

The CL-CSEN framework is illustrated in Fig. 5 where the mapping from low-dimensional to high-dimensional space is learned during training. In this way, the proxy mapping layer is jointly optimized with the CSEN part of the CL-CSEN model to maximize the regression performance. Hence, the input and output pair of the proposed method with CL-CSEN will be $\left(\mathbf{y}^{\text{train}}, d^{\text{train}}\right)$ for the training.

4. Experimental evaluation

The performance of the proposed approach is evaluated over the KITTI 3D Object Detection (Geiger et al., 2012) dataset. KITTI provides 3D bounding boxes for the detected objects as well as their categories. Besides having 3D object dimensions including length, height, and width, the dataset has the information of the 3D object locations: x,y, and z in camera coordinates. Hence, we use the z location information as the ground truth for the object distance estimation task. The collected frames are captured by a moving platform/vehicle from rural areas, a mid-size city, and highways. One challenge with this dataset is that there are overlapping samples on the observed scene as illustrated in Fig. 6.

4.1. Experimental setup

The KITTI annotations consist of 7481 images and there are a total of 40 570 objects having the distance information. The majority of them, 38 307 objects are in the range of [0.5, 60.5] meters. In this study, the objects between the given range are selected for the evaluation in order to remove the outlier objects that are significantly far or close to the camera. The selected and cropped objects are then resized to 64×64 images and fed to the different feature extractor networks. We have created two different experimental setups. In the first one, a total of 19 769 samples are randomly selected for the training split and the remaining 18 538 samples are for the testing. In the second, only 4800 samples are used for training while the majority (33 507 samples) are used for the test. Consequently, these scenarios fulfill the aim of this study, i.e., evaluation of the learning capability with the limited amount of data (approximately 50% and less than 13% of the annotated data in the first and latter scenarios, respectively).

4.1.1. CSEN and CL-CSEN configurations

To form the dictionary **D**, we allocate 1200 samples from the training split and quantize those samples using 61 partitions in such a way that at the end, there are 20 samples per meter ($20 \times 60 = 1200$ objects in total) within the selected distance range. Recall the fact that these selected samples for each meter consist of different object categories such as person, car, truck, and trailer. Thus, **D** consists of 1200 samples in the proposed approaches with the CSEN and CL-CSEN. The compression ratio is set to CR $= m/d = 0.5$ using the PCA matrix **A** that is computed using the allocated samples for the dictionary formation. Consequently, after the compression is applied, the size of the equivalent dictionary **D** would be $m \times 1200$ where $m = 512, 256$, and 1024 for DenseNet-121, VGG19, and ResNet-50, respectively. Consequently, the corresponding denoiser matrix $\mathbf{B} = \left(\mathbf{D}^T\mathbf{D} + \lambda\mathbf{I}\right)^{-1}\mathbf{D}^T$ would be $1200 \times m$. Therefore, the reshaped version of the computed proxy signal $\tilde{\mathbf{x}} = \mathbf{B}\mathbf{y}$, $\tilde{\mathbf{x}} \in \mathbb{R}^{n=1200}$ has the size of 80×15 in the 2-D plane. Finally, the remaining training samples are used for the training of CSEN and CL-CSEN.

The proposed compact CSEN structure given in Fig. 3 consists of only two convolutional (both with 5×5 filter sizes) and one dense layer. The first convolutional layer has 64 weight kernels that is followed by max-pooling with 4×5 pooling size. The second convolutional layer has only one kernel that creates a feature map that is flattened and connected to the single output neuron. In the CL-CSEN, there are additional two fully connected dense layers with the number of neurons corresponding to the size of $\mathbf{B}^T$ as previously discussed. In this way, the followed compact structure brings the ability to learn from a limited amount of
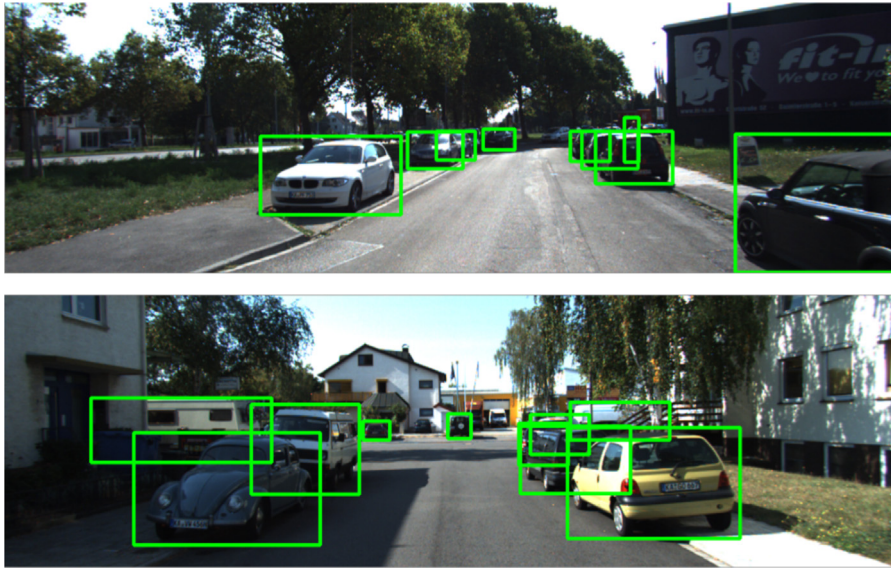
Fig. 6. Two sample frames from KITTI 3D Object Detection showing overlapped samples.

data. All the layers have ReLu as the activation function except the output that has the SoftPlus activation function.

The CSEN is trained with 100 epochs and batch size of 16 by Adam optimizer (Kingma & Ba, 2014) using the proposed default parameter values as a learning rate $\alpha = 10^{-3}$, $\beta_1 = 0.9$, and $\beta_2 = 0.999$. From the training set, we separate 20% of samples for the validation to select the best network model to be used for testing. The experiments have been performed using Python on a PC with NVidia$^{\circledR}$ 1080 Ti GPU card, Intel$^{\circledR}$ i9 − 7900X CPU having 128 GB system memory. The CSEN and CL-CSEN are implemented with the Tensorflow library (Abadi et al., 2016). The hyper-parameter of λ is first searched in log-scale within the range $\lambda^* \in [10^{-13}, 10^3]$. Afterwards, the fine-tuned version is set with few more steps by slight adjustment such that $\lambda = \lambda^* \pm 10^{log(\lambda^*)}$.

4.1.2. Competing methods

Since we propose to use RbR by utilizing the regularized least-square sense solution as the coarse estimation of the support sets, the performance analysis will be performed against the base model with CRC (Zhang et al., 2011), and then, the improvement over CRC by the proposed CSEN and CL-CSEN will be reported. Moreover, we use various different solvers for SRC approach including ADMM (Boyd, Parikh, Chu, Peleato, Eckstein, et al., 2011), Primal and Dual Augmented Lagrangian Methods, (PALM and DALM) (Yang, Zhou, Balasubramanian, Sastry, & Ma, 2013), Orthogonal Matching Pursuit (OMP) (Yang et al., 2013), Homotopy (Malioutov, Cetin, & Willsky, 2005), Gradient Projection for Sparse Reconstruction (GPSR) (Figueiredo, Nowak, & Wright, 2007), ℓ_1 regularized Least Squares (ℓ_1-LS) (Koh, Kim, & Boyd, 2007), ℓ_1-magic (Candes & Romberg, 2005). Next, the following two representation based classification approaches are also included in the evaluation: Superposed Linear Representation Classifier (SLRC) (Deng, Hu, & Guo, 2018) and Nonnegative Representation-based Classifier (NRC) (Xu, An, Zhang, & Zhang, 2019). The SLRC approach contains both ℓ_1 and ℓ_2 minimization depending on the selected regularization of the coding coefficients. In this study, SLRC-ℓ_1 version is used for the comparisons. Additionally, the SVR approach that has been used by Gökçe et al. (2015) and Zhu and Fang (2019) for distance estimation is included in comparisons. Note that compared to Gökçe et al. (2015) and Zhu and Fang (2019), we use the enhanced features obtained by the feature extraction method explained earlier.

The SVR configuration is developed by searching the optimal hyper-parameters. Accordingly, the grid-search is applied over the validation set with the following kernel functions: linear, Radial Basis Function (RBF), and polynomial using the following parameters: γ parameter (kernel coefficients for the RBF and polynomial kernels) in the range $[10^{-3}, 10^3]$ by varying in the log-scale, the degree of the polynomial $\{2, 3, 4\}$, the regularization parameter (C parameter) in the range $[10^{-3}, 10^3]$ by varying in the log-scale.

To make a fair comparison with the competing methods, the training set of SVR includes also the dictionary samples in addition to the training samples that are used in the proposed CSEN and CL-CSEN. Similarly, the dictionary samples in SRC and CRC methods include the training samples plus the dictionary samples of CSEN and CL-CSEN. The same feature extraction procedure in the proposed method is used in the SRC, CRC, and SVR (i.e., ϕ ($\mathbf{I}_i$) where ϕ is the pre-trained network for the cropped and resized object $\mathbf{I}_i$). The same CR is used by the PCA for SRC and CRC. In SVR, it is not feasible to compute the exact solution due to the scale of the data; and hence, we use Nystroem method (Williams & Seeger, 2001; Yang, Li, Mahdavi, Jin, & Zhou, 2012) for the kernel approximation in order to approximate $m = CR \times d$ number of feature maps where CR = 0.5. Overall, we keep the same CR value for all the methods in the experimental evaluations.

4.2. Experimental results

The same performance metrics as used in many studies (Casser et al., 2019; Chang & Chen, 2018; Haseeb et al., 2018; Mahjourian et al., 2018; Wang et al., 2019; Zhu & Fang, 2019) are used to evaluate the distance estimation performance of the proposed approach including Absolute Relative Distance (ARD), Squared Relative Distance (SRD), Root of Mean Squared Error (RMSE), the root of Mean Squared logarithmic Error (RMSE$_{log}$), and thresholded ratio between the ground-truth and prediction.

The distance estimation performance of the proposed method is presented in Table 1. In these results, we report the RbR performance with the proposed CSEN and CL-CSEN models where the quantization is not applied for the training and testing samples, but is only used for the dictionary reconstruction. Correspondingly, the train:test splits are chosen as approximately 1:1 proportion. Note the fact that multiple pre-trained networks are utilized for feature extraction. In this way, we aim to evaluate the

Table 1

The statistical (mean and standard deviations) performance metrics are reported from five different runs to show the object distance estimation performance of the proposed approach against the competing methods over the KITTI dataset and using different feature extractor networks, $\phi : \mathbb{R}^{N \times N \times 3} \to \mathbb{R}^d$. The train:test splits are selected as approximately 1:1 proportion. In the metrics, ↓:lower is better and ↑: higher is better.

$\phi(\cdot)$	Method	ARD ↓	SRD ↓	RMSE ↓	RMSE$_{log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
DenseNet-121	Support Vector Regressor (SVR) (Gökçe et al., 2015)	0.2588 ± 0.003	1.7764 ± 0.041	5.3239 ± 0.013	0.4189 ± 0.006	0.6908 ± 0.002	0.8862 ± 0.003	0.9433 ± 0.003
	Base Model (CRC-light) (Zhang et al., 2011)	0.4183 ± 0.008	6.9585 ± 0.346	12.0007 ± 0.164	0.7462 ± 0.014	0.4447 ± 0.002	0.6821 ± 0.003	0.8055 ± 0.003
	CSEN (Proposed)	0.2828 ± 0.006	2.2385 ± 0.091	6.2951 ± 0.061	0.4344 ± 0.052	0.6268 ± 0.008	0.8630 ± 0.003	0.9367 ± 0.002
	CL-CSEN (Proposed)	**0.2005 ± 0.009**	**1.2137 ± 0.084**	**4.3413 ± 0.048**	**0.2720 ± 0.014**	**0.7870 ± 0.006**	**0.9361 ± 0.005**	**0.9704 ± 0.003**
VGG19	Support Vector Regressor (SVR) (Gökçe et al., 2015)	0.3496 ± 0.007	3.1122 ± 0.131	6.9459 ± 0.045	0.4690 ± 0.023	0.5752 ± 0.006	0.8325 ± 0.004	0.9172 ± 0.001
	Base Model (CRC-light) (Zhang et al., 2011)	0.4029 ± 0.003	6.1675 ± 0.134	12.2411 ± 0.112	0.8556 ± 0.031	0.4266 ± 0.004	0.6492 ± 0.006	0.7682 ± 0.007
	CSEN (Proposed)	0.2917 ± 0.013	2.3542 ± 0.150	6.4498 ± 0.037	0.4581 ± 0.081	0.6058 ± 0.006	0.8510 ± 0.009	0.9307 ± 0.008
	CL-CSEN (Proposed)	**0.2221 ± 0.010**	**1.5034 ± 0.116**	**4.8132 ± 0.044**	**0.3021 ± 0.012**	**0.7448 ± 0.006**	**0.9164 ± 0.004**	**0.9623 ± 0.003**
ResNet-50	Support Vector Regressor (SVR) (Gökçe et al., 2015)	0.2509 ± 0.004	1.7669 ± 0.052	5.3531 ± 0.049	0.3613 ± 0.004	0.7004 ± 0.004	0.8989 ± 0.002	0.9519 ± 0.001
	Base Model (CRC-light) (Zhang et al., 2011)	0.3781 ± 0.004	5.6263 ± 0.146	10.9210 ± 0.082	0.6472 ± 0.011	0.4712 ± 0.002	0.7193 ± 0.001	0.8443 ± 0.002
	CSEN (Proposed)	0.2400 ± 0.006	1.6777 ± 0.073	5.5212 ± 0.087	0.3459 ± 0.011	0.6902 ± 0.008	0.8983 ± 0.004	0.9533 ± 0.002
	CL-CSEN (Proposed)	**0.1934 ± 0.009**	**1.1710 ± 0.097**	**4.0849 ± 0.044**	**0.2604 ± 0.008**	**0.8148 ± 0.005**	**0.9439 ± 0.004**	**0.9730 ± 0.002**

(a) DenseNet-121 Feature Extraction

(b) VGG-19 Feature Extraction.
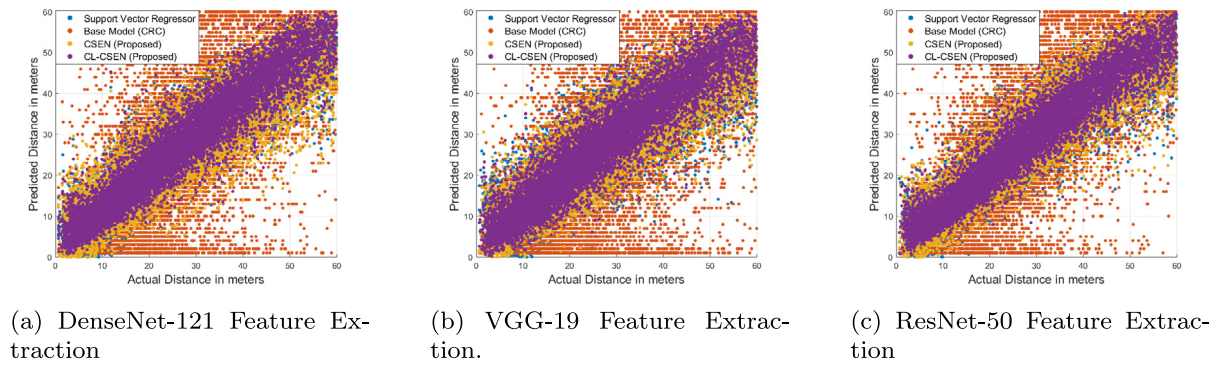
(c) ResNet-50 Feature Extraction

Fig. 7. Predicted vs. actual distances of the objects in the test set for the proposed CSEN and CL-CSEN and compared methods using different feature extractor networks. In the scattering plot, each point represents a sample object in the KITTI dataset that is partitioned to train:test corresponding approximately 1:1 proportion.
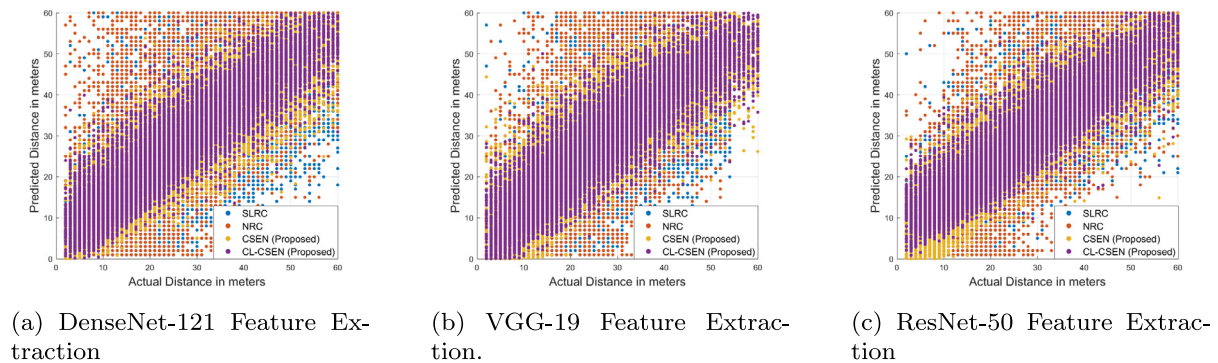

(a) DenseNet-121 Feature Extraction

(b) VGG-19 Feature Extraction.

(c) ResNet-50 Feature Extraction

Fig. 8. Predicted vs. actual distances of the objects in the test set for the proposed CSEN and CL-CSEN and compared methods using different feature extractor networks. In the scattering plot, each point represents a sample object in the KITTI dataset that is partitioned to train:test corresponding approximately 1:17 proportion. The selected distance sensitivity (with quantization) is one meter.

performance effect of different network architectures in feature extraction. DenseNet-121 has skip-connections that connect each layer to every other layer so that each layer is densely connected, ResNet-50 only has skip-connections between every second layer, and VGG-19 does not have any such shortcut connection between the layers. Based on Table 1, a higher estimation accuracy is achieved by the proposed approach compared to SVR and the performance is highly improved compared to our base CRC model. Moreover, the proposed method outperforms (Zhu & Fang, 2019) even though they use additional information such as the categorical class information of the objects and the projection matrix for the training. For a more fair comparison, the proposed method is also compared with the base model of Zhu and Fang (2019) without classification; and the performance gap becomes even higher as expected. Additionally, scattering plots are provided in Fig. 7 demonstrating the actual distance versus the predicted distance by all methods. Correspondingly, we expect to see an identity transformation ideally. In the plots, the sample point sizes are purposely selected bigger to better illustrate the misdetections. Thus, considering the number of samples, most of them located at the identity line region and give a constant color view. Hence, it is observed that the CL-CSEN method provides the least scattered samples compared to the other methods especially when ResNet-50 features are used. Even though the reported metrics in Table 1 show improvements achieved by the proposed method; the performance gain is more visible for distant objects considering that the gap is significant in the squared metrics.

Next, the performance comparison is provided in Table 2 regarding the proposed method with CSEN and CL-CSEN versus other representation-based classification approaches. In this set of experiments, contrary to Table 1, we have applied quantization to the training and testing samples of the CSEN and CL-CSEN

approaches as previously discussed. Even though they can be trained for the full regression task, the competing representation-based classification methods do not have this ability; and hence, we wanted to compare the proposed approach fairly with them. In the table, CRC-light corresponds to our coarse estimation for the CSEN and CL-CSEN methods where the same number of dictionary samples are used in the CSEN approach. Based on Table 2, it is clear that both proposed methods have achieved a significant performance gap over the competing methods. It is also shown that the CSEN and CL-CSEN methods are able to learn from such a limited number of training samples (only 4800 samples are used for training compared to 33 507 testing samples). Similarly, the scattering plot is provided in Fig. 8 for the second set of the experiments and the best-performing methods from Table 2. Based on the plots, CL-CSEN has less distributed scatters due to the improved distance estimation performance: most of the test samples are overlapped near the identity transformation where few test samples are separated or distinguishable from the others since they are detected in error and located far from the overlapped points. The visual difference from the previous plot in Fig. 7 is that expectedly, the samples are located with 1 m distances due to the applied quantization.

Three sample frames are shown in Fig. 9 with their corresponding true and the estimated object-specific distances by the three best-performing methods: SVR, CSEN, and CL-CSEN. The first frame (first column in Fig. 9) represents a typical sample from the KITTI 3D Object Detection dataset in which there are overlapping objects. Moreover, even though the dataset may have some well-separated samples, the illumination conditions make it harder to perform analysis as observed in the third sample frame in Fig. 9. Visual inspection based on these frames indicates that even though CL-CSEN provides enhanced performance than CSEN

Table 2
The statistical (mean and standard deviations) performance metrics are reported from five different runs to show the object distance estimation performance of the proposed approach against the competing methods over the KITTI dataset and using differenta feature extractor networks, $\phi : \mathbb{R}^{N \times N \times 3} \rightarrow \mathbb{R}^d$. The train:test splits are selected as approximately 1:17 proportion and the selected distance sensitivity (with quantization) is 1 m. In the metrics, ↓:lower is better and ↑: higher is better.

| $\phi(\cdot)$	Method	ARD ↓	SRD ↓	RMSE ↓	RMSE$_{log}$ ↓	$\delta < 1.25$ ↑	$\delta < 1.25^2$ ↑	$\delta < 1.25^3$ ↑
DenseNet-121	CRC-light (Zhang et al., 2011)	0.4157 ± 0.0.10	6.9163 ± 0.362	12.0034 ± 0.159	0.7442 ± 0.010	0.4337 ± 0.002	0.6802 ± 0.004	0.8018 ± 0.003
	CRC (Zhang et al., 2011)	0.3384 ± 0.003	4.9194 ± 0.110	11.1735 ± 0.068	0.9687 ± 0.007	0.5060 ± 0.002	0.7091 ± 0.003	0.7887 ± 0.002
	ADMM (Boyd et al., 2011)	0.3662 ± 0.001	5.7964 ± 0.109	9.7512 ± 0.092	0.5286 ± 0.004	0.5429 ± 0.002	0.7731 ± 0.002	0.8737 ± 0.001
	DALM (Yang et al., 2013)	0.3502 ± 0.003	5.3903 ± 0.108	9.7558 ± 0.107	0.5566 ± 0.006	0.5498 ± 0.003	0.7753 ± 0.003	0.8706 ± 0.002
	OMP (Yang et al., 2013)	0.4279 ± 0.004	8.0630 ± 0.142	11.1767 ± 0.072	0.6326 ± 0.005	0.5203 ± 0.003	0.7386 ± 0.003	0.9019 ± 0.002
	Homotopy (Malioutov et al., 2005)	0.3747 ± 0.004	5.8120 ± 0.075	9.5982 ± 0.053	0.4917 ± 0.003	0.5415 ± 0.003	0.7764 ± 0.003	0.8806 ± 0.002
	GPSR (Figueiredo et al., 2007)	0.3357 ± 0.003	4.9460 ± 0.099	9.4322 ± 0.089	0.5456 ± 0.003	0.5547 ± 0.001	0.7824 ± 0.001	0.8777 ± 0.001
	ℓ_1-LS (Koh et al., 2007)	0.3550 ± 0.003	5.4403 ± 0.096	11.9869 ± 0.090	1.0916 ± 0.010	0.4932 ± 0.002	0.6840 ± 0.004	0.7584 ± 0.004
	ℓ_1-magic (Candes & Romberg, 2005)	0.3579 ± 0.004	5.6748 ± 0.141	9.9567 ± 0.114	0.5695 ± 0.006	0.5457 ± 0.003	0.7695 ± 0.002	0.8655 ± 0.001
	PALM (Yang et al., 2013)	0.3262 ± 0.003	4.6281 ± 0.109	10.6396 ± 0.075	0.8858 ± 0.006	0.5278 ± 0.003	0.6919 ± 0.082	0.8117 ± 0.002
	SLRC (Deng et al., 2018)	**0.3144 ± 0.003**	3.6487 ± 0.064	9.6575 ± 0.099	0.5511 ± 0.005	0.4094 ± 0.004	0.6935 ± 0.007	0.8440 ± 0.006
	NRC (Xu et al., 2019)	0.3356 ± 0.005	5.0012 ± 0.184	9.4777 ± 0.095	0.5260 ± 0.004	0.5581 ± 0.002	0.7881 ± 0.004	0.8826 ± 0.001
	CSEN (Proposed)	0.3308 ± 0.036	3.0487 ± 0.534	6.9950 ± 0.367	0.5370 ± 0.217	0.5668 ± 0.023	0.8215 ± 0.025	0.9132 ± 0.020
	CL-CSEN (Proposed)	0.3167 ± 0.006	**2.8268 ± 0.083**	**6.2036 ± 0.046**	**0.3753 ± 0.017**	**0.6378 ± 0.004**	**0.8555 ± 0.004**	**0.9313 ± 0.002**
VGG19	CRC-light (Zhang et al., 2011)	0.4018 ± 0.002	6.1850 ± 0.083	12.2543 ± 0.101	0.8549 ± 0.032	0.4163 ± 0.004	0.6498 ± 0.005	0.7664 ± 0.007
	CRC (Zhang et al., 2011)	0.3591 ± 0.004	5.3996 ± 0.057	12.1625 ± 0.075	1.0796 ± 0.016	0.4727 ± 0.002	0.6576 ± 0.006	0.7333 ± 0.008
	ADMM (Boyd et al., 2011)	0.3506 ± 0.005	5.3399 ± 0.138	9.4499 ± 0.066	0.5114 ± 0.005	0.5547 ± 0.001	0.7829 ± 0.003	0.8799 ± 0.001
	DALM (Yang et al., 2013)	0.3535 ± 0.004	5.4561 ± 0.129	9.8062 ± 0.062	0.5653 ± 0.006	0.5466 ± 0.002	0.7697 ± 0.002	0.8656 ± 0.002
	OMP (Yang et al., 2013)	0.3946 ± 0.004	6.7427 ± 0.119	10.2869 ± 0.028	0.5589 ± 0.004	0.5395 ± 0.003	0.7621 ± 0.002	0.8598 ± 0.001
	Homotopy (Malioutov et al., 2005)	0.3532 ± 0.005	5.2429 ± 0.186	9.1591 ± 0.039	0.4624 ± 0.002	0.5604 ± 0.002	0.7931 ± 0.002	0.8921 ± 0.001
	GPSR (Figueiredo et al., 2007)	0.3301 ± 0.003	4.7034 ± 0.108	9.3988 ± 0.062	0.5679 ± 0.007	0.5540 ± 0.002	0.7773 ± 0.002	0.8711 ± 0.001
	ℓ_1-LS (Koh et al., 2007)	0.3683 ± 0.005	5.6823 ± 0.077	12.5880 ± 0.092	1.1247 ± 0.018	0.4641 ± 0.003	0.6432 ± 0.007	0.7166 ± 0.009
	ℓ_1-magic (Candes & Romberg, 2005)	0.3541 ± 0.004	5.4787 ± 0.118	9.8251 ± 0.058	0.5669 ± 0.006	0.5464 ± 0.002	0.7692 ± 0.002	0.8652 ± 0.001
	PALM (Yang et al., 2013)	0.3175 ± 0.001	4.2531 ± 0.026	10.3851 ± 0.041	0.8573 ± 0.009	0.5261 ± 0.001	0.7267 ± 0.003	0.8059 ± 0.005
	SLRC (Deng et al., 2018)	0.3089 ± 0.002	3.6981 ± 0.101	9.2953 ± 0.033	0.5066 ± 0.005	0.4573 ± 0.005	0.7372 ± 0.003	0.8714 ± 0.002
	NRC (Xu et al., 2019)	0.3280 ± 0.002	4.6455 ± 0.058	9.3277 ± 0.049	0.5190 ± 0.005	0.5584 ± 0.002	0.7870 ± 0.002	0.8836 ± 0.001
	CSEN (Proposed)	0.3401 ± 0.039	3.1667 ± 0.563	7.2027 ± 0.331	0.6763 ± 0.264	0.5392 ± 0.019	0.7978 ± 0.021	0.9000 ± 0.017
	CL-CSEN (Proposed)	**0.3062 ± 0.010**	**2.6452 ± 0.140**	6.3759 ± 0.122	**0.4222 ± 0.059**	**0.6091 ± 0.009**	**0.8404 ± 0.009**	**0.9265 ± 0.006**
ResNet-50	CRC-light (Zhang et al., 2011)	0.3752 ± 0.003	5.5853 ± 0.081	10.8963 ± 0.066	0.6454 ± 0.014	0.4605 ± 0.001	0.7184 ± 0.003	0.8410 ± 0.002
	CRC (Zhang et al., 2011)	0.2817 ± 0.002	3.3945 ± 0.063	9.1777 ± 0.080	0.7371 ± 0.013	0.5598 ± 0.003	0.7786 ± 0.005	0.8562 ± 0.004
	ADMM (Boyd et al., 2011)	0.3155 ± 0.003	4.2173 ± 0.062	8.6938 ± 0.062	0.4798 ± 0.006	0.5680 ± 0.002	0.8038 ± 0.002	0.8979 ± 0.002
	DALM (Yang et al., 2013)	0.2916 ± 0.003	3.6398 ± 0.075	8.4626 ± 0.082	0.4981 ± 0.008	0.5791 ± 0.003	0.8128 ± 0.004	0.9019 ± 0.002
	OMP (Yang et al., 2013)	0.3352 ± 0.003	4.9965 ± 0.089	9.5783 ± 0.067	0.5639 ± 0.005	0.5550 ± 0.003	0.7813 ± 0.004	0.8761 ± 0.002
	Homotopy (Malioutov et al., 2005)	0.3239 ± 0.005	4.2413 ± 0.103	8.4395 ± 0.026	0.4424 ± 0.004	0.5711 ± 0.001	0.8098 ± 0.002	0.9069 ± 0.002
	GPSR (Figueiredo et al., 2007)	0.2928 ± 0.003	3.6532 ± 0.094	8.4384 ± 0.100	0.4963 ± 0.007	0.5791 ± 0.003	0.8127 ± 0.003	0.9012 ± 0.002
	ℓ_1-LS (Koh et al., 2007)	0.2849 ± 0.004	3.4606 ± 0.090	9.4654 ± 0.078	0.7684 ± 0.011	0.5540 ± 0.003	0.7657 ± 0.005	0.8419 ± 0.004
	ℓ_1-magic (Candes & Romberg, 2005)	0.2940 ± 0.002	3.6942 ± 0.057	8.5528 ± 0.059	0.5061 ± 0.007	0.5758 ± 0.002	0.8083 ± 0.003	0.8979 ± 0.002
	PALM (Yang et al., 2013)	0.2767 ± 0.003	3.2185 ± 0.090	8.9170 ± 0.088	0.6784 ± 0.012	0.5668 ± 0.003	0.7855 ± 0.004	0.8643 ± 0.004
	SLRC (Deng et al., 2018)	**0.2431 ± 0.001**	2.3720 ± 0.037	7.4747 ± 0.034	0.3850 ± 0.006	0.5718 ± 0.003	0.8476 ± 0.003	**0.9378 ± 0.002**
	NRC (Xu et al., 2019)	0.2729 ± 0.003	3.1236 ± 0.058	8.0496 ± 0.064	0.4742 ± 0.009	0.5932 ± 0.003	0.8278 ± 0.003	0.9136 ± 0.002
	CSEN (Proposed)	0.2835 ± 0.035	**2.2479 ± 0.437**	6.2142 ± 0.333	0.5074 ± 0.155	0.6076 ± 0.020	0.8481 ± 0.016	0.9280 ± 0.011
	CL-CSEN (Proposed)	0.3359 ± 0.010	2.9720 ± 0.106	**6.0765 ± 0.041**	**0.3735 ± 0.005**	**0.6398 ± 0.005**	**0.8510 ± 0.004**	0.9261 ± 0.003
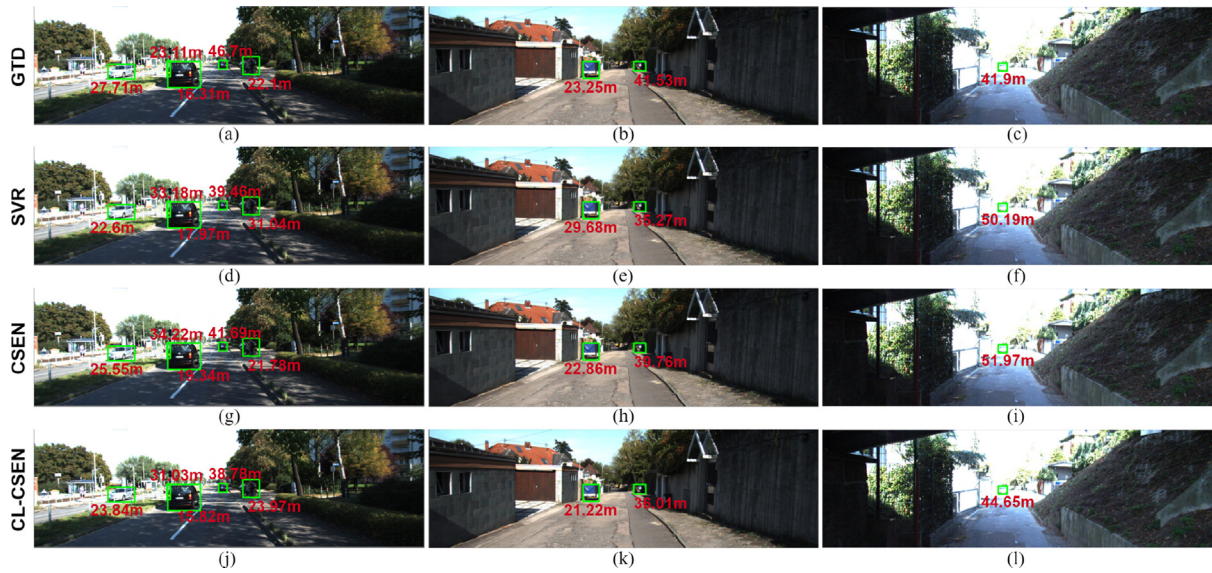
Fig. 9. Three sample frames are shown with the object bounding boxes and their corresponding ground-truth distances (GTD) in the first row: (a), (b), and (c). Then, the estimated distances for the objects by the three best-performing methods in this work: SVR, CSEN, and CL-CSEN are illustrated in the second (d - e - f), third (g - h - i), and the last (j - k - l) rows, respectively. The approximate 1:1 ratio is followed in train:test splits.

Table 3
The number of trainable parameters is given in (a) for the proposed CSEN and CL-CSEN models. The elapsed times using the aforementioned PC setup are given in (b) for the methods.

(a) The trainable parameters using different feature extractors ϕ.

$\phi : \mathbb{R}^{N \times N \times 3} \to \mathbb{R}^d$	Model	Number of parameters
$d \in \{1024, 512, 2048\}$	CSEN	3,326
$d = 1024$	CL-CSEN	618,926
$d = 512$	CL-CSEN	311,726
$d = 2048$	CL-CSEN	1,233,326

(b) Average elapsed times in milliseconds (ms) for the estimation of a test object sample. The given computational times are obtained in the case of ResNet-50 features.

Method	Time (ms)
SVR	0.0035
CRC-light	2.0242
CRC	14.258
ADMM	198.14
DALM	3574.0
OMP	241.22
Homotopy	30.591
GPSR	1547.0
ℓ_1-LS	223.84
ℓ_1-magic	2698.2
PALM	10996.0
CSEN	0.0348
CL-CSEN	0.0320
SLRC	678.2256
NRC	127.8132

according to the quantitative analysis, for the close objects, CSEN seems to provide more accurate results. However, CSEN starts to underperform compared to CL-CSEN when the objects are distant from the camera.

4.3. Computational complexity analysis

The number of trainable parameters is provided in Table 3a for the proposed CSEN and CL-CSEN models. Accordingly, the CSEN model has only a few thousand trainable parameters since the denoiser matrix **B** is not trainable, whereas in the CL-CSEN

model, depending on the size of **B** the trainable parameters vary. Nevertheless, both are still compact architectures only with a few layers. The elapsed times are reported in Table 3b on the aforementioned PC setup. On the other hand, SRC methods suffer drastic time complexity whereas elapsed times for CSEN and CL-CSEN methods are comparable with the SVR method. Note the fact that even though the CL-CSEN pipeline has more trainable parameters, the required time for the inference is less than CSEN. Because; the proxy mapping and reshaping stages for the following convolutional layers are implemented on GPU as an end-to-end pipeline that brings the computational efficiency that was lacking in the initial CSEN approach. Note that even though CSEN and CL-CSEN utilize the proxy mapping stage of the CRC approach, they are still computationally efficient because CRC-light and CRC require additional residual finding step that was explained in Section 2.2.

4.4. Discussion: 1-D versus 2-D proxy signal representation

The presented distance estimation results are obtained using the proposed CSEN and CL-CSEN approaches that contain 2-D convolution operations. One can investigate operating directly over the 1-D proxy signal without further reshaping it as the input of the first convolutional layer. Hence, we present the distance estimation results in Table 4 using 1-D convolutional layers in the proposed approaches. Accordingly, the CSEN-1D and CL-CSEN-1D models do not have any reshaping operations contrary to 2-D versions illustrated in Figs. 3 and 5. It is observed that using the same number of trainable parameters, i.e., 25×1 filter sizes for each convolutional layer, the comparable results are obtained by performing 1-D inference on the proxy signal.

5. Limitations of the study

There are two possible feature extraction procedures that can be considered when building the representative dictionary: i. traditional hand-crafted feature extraction such as PCA and using bounding box dimensions as features and ii. extracting learned features. If the first category features are used in the dictionary, there is no requirement for prior resizing and one can select the first d principal components for PCA features, e.g., $d \in$

Table 4
The statistical (mean and standard deviations) performance metrics are reported from five different runs using the 1D versions of the proposed approaches (CSEN-1D and CL-CSEN-1D) over the KITTI dataset and using different feature extractor networks, $\phi : \mathbb{R}^{N \times N \times 3} \rightarrow \mathbb{R}^d$. In the metrics, ↓:lower is better and ↑: higher is better.

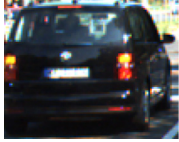(a) The train:test splits are selected as approximately 1:1 proportion.

$\phi(\cdot)$	DenseNet-121		VGG19		ResNet-50	
	CSEN-1D	CL-CSEN-1D	CSEN-1D	CL-CSEN-1D	CSEN-1D	CL-CSEN-1D
ARD ↓	0.3000 ± 0.013	0.2092 ± 0.005	0.3071 ± 0.011	0.2289 ± 0.004	0.2507 ± 0.006	0.1978 ± 0.005
SRD ↓	2.4741 ± 0.155	1.3051 ± 0.059	2.5457 ± 0.129	1.5757 ± 0.058	1.7831 ± 0.077	1.2107 ± 0.060
RMSE ↓	6.3539 ± 0.029	4.4239 ± 0.024	6.5480 ± 0.049	4.8954 ± 0.026	5.5808 ± 0.037	4.1575 ± 0.027
RMSE$_{log}$ ↓	0.5490 ± 0.092	0.2885 ± 0.006	0.5717 ± 0.064	0.3083 ± 0.009	0.4283 ± 0.060	0.2660 ± 0.004
$\delta < 1.25$ ↑	0.6182 ± 0.006	0.7765 ± 0.004	0.5957 ± 0.004	0.7364 ± 0.001	0.6762 ± 0.009	0.8087 ± 0.002
$\delta < 1.25^2$ ↑	0.8492 ± 0.006	0.9297 ± 0.002	0.8375 ± 0.007	0.9111 ± 0.001	0.8845 ± 0.008	0.9414 ± 0.002
$\delta < 1.25^3$ ↑	0.9251 ± 0.006	0.9672 ± 0.002	0.9199 ± 0.006	0.9602 ± 0.002	0.9432 ± 0.006	0.9716 ± 0.001

(b) The train:test splits are selected as approximately 1:17 proportion and the distance sensitivity (with quantization) is 1 m.

$\phi(\cdot)$	DenseNet-121		VGG19		ResNet-50	
	CSEN-1D	CL-CSEN-1D	CSEN-1D	CL-CSEN-1D	CSEN-1D	CL-CSEN-1D
ARD ↓	0.3365 ± 0.023	0.3457 ± 0.018	0.3365 ± 0.033	0.3195 ± 0.012	0.2923 ± 0.017	0.3608 ± 0.017
SRD ↓	3.0839 ± 0.351	3.3374 ± 0.293	3.0596 ± 0.492	2.8948 ± 0.212	2.3585 ± 0.227	3.3650 ± 0.225
RMSE ↓	7.0004 ± 0.207	6.5406 ± 0.144	7.0975 ± 0.293	6.6121 ± 0.209	6.2865 ± 0.234	6.3010 ± 0.079
RMSE$_{log}$ ↓	0.7130 ± 0.193	0.3893 ± 0.007	0.8101 ± 0.201	0.4091 ± 0.054	0.6495 ± 0.210	0.3913 ± 0.011
$\delta < 1.25$ ↑	0.5573 ± 0.012	0.6219 ± 0.005	0.5408 ± 0.015	0.5944 ± 0.008	0.6056 ± 0.014	0.6273 ± 0.006
$\delta < 1.25^2$ ↑	0.8064 ± 0.016	0.8417 ± 0.005	0.7930 ± 0.014	0.8323 ± 0.004	0.8478 ± 0.012	0.8385 ± 0.007
$\delta < 1.25^3$ ↑	0.9003 ± 0.015	0.9221 ± 0.004	0.8930 ± 0.013	0.9230 ± 0.004	0.9247 ± 0.013	0.9172 ± 0.006

Table 5
Objects (o_1 - o_5) are given with their corresponding distances after cropping and resizing operations. The ARD errors are calculated for SVR, CSEN, and CL-CSEN approaches.

	o_1: 16.31 m	o_2: 22.1 m	o_3: 23.25 m	o_4: 41.53 m	o_5: 41.9 m
ARD	SVR: 0.102	SVR: 0.405	SVR: 0.277	SVR: 0.151	SVR: 0.198
	CSEN: 0.186	CSEN: 0.015	CSEN: 0.017	CSEN: 0.043	CSEN: 0.240
	CL-CSEN: 0.030	CL-CSEN: 0.085	CL-CSEN: 0.087	CL-CSEN: 0.133	CL-CSEN: 0.066

{1024, 512, 2048} and stack them column-wise. A similar feature extraction procedure is followed by Wright et al. (2010, 2008), Zhang et al. (2011), e.g., in gray-level face recognition. The bounding box dimension has been used by Haseeb et al. (2018) in the distance estimation problem. The corresponding drawback is that it requires the classification of the objects. Because, the dimensions of different object categories are different, e.g., truck versus pedestrian. Obviously, the distance estimation accuracy depends on the classification performance and in practice, it is unfeasible to define a sufficient number of object categories. For example, even if the object is correctly classified as a truck, a distance estimation algorithm using the bounding box dimensions would likely fail in measuring the distance of a small truck (having a small bounding box) and a large truck with a large bounding box.

In this study, we found out that when the dictionary **D** is constructed by PCA features or a bounding box dimension with an additional classification framework, the methods were unable to provide any reasonable performance and most often the convergence did not happen. We observe that more descriptive features are needed since the task is more challenging compared to the aforementioned studies. Therefore, we propose to use

learned features extracted by DenseNet-121, VGG19, and ResNet-50. The limitation of the pre-trained feature extractors is that they need fixed-size input images and as discussed in Section 3.1 and illustrated in Fig. 2, the cropped objects are therefore resized to a fixed-size in order to feed them into the networks. We limit the effect of the resizing operation by using a simple bilinear interpolation. For instance, it is shown in Fig. 2 and Table 5 (e.g., o_4 and o_5) that up-sampling does not produce sharper images for distant objects. In the previous discussion, we mentioned that the blurry appearance of the objects can be correlated with the distance and the distant objects tend to have blurry appearances after resizing. One can argue that blurry appearances might occur also in closer objects if they are captured already in a blur. Therefore, in the experimentation of this study, we have observed that objects are either captured with no blur or with the relative blurry appearances due to motion or other factors, that have been learned during the training since the camera has a fixed focus. It is also observed that the CL-CSEN method is more robust to blurriness than other methods. For example, in Table 5, o_4 and o_5 have approximately the same distances to the camera, but the latter has a blurrier appearance. A similar observation can be

made for o_2 and o_3 where the blurriness is not equal between the two objects even though they have similar distances. In both cases, it can be seen that CL-CSEN is able to produce small ARD errors comparable to o_1.

6. Conclusion

In this study, we first propose a novel CSEN-based distance estimation method using a single camera. CSENs were recently proposed to directly estimate support sets of a signal instead of the traditional approach, i.e., first reconstructing the sparse signal and applying a threshold. Using the modified CSENs for regression, we demonstrate that it is possible to utilize representative dictionaries for a regression task; and to the best of authors' knowledge, this makes the pioneer study in this domain. Hence, we introduce the term *Representation-based Regression (RbR)* to reflect this fact. Similar to SRC and CRC methods, the introduced *RbR* approach is able to operate over scarce data. Moreover, utilizing the introduced representative dictionary design by collecting the samples with the same distances in the quantization level, the performance of the proposed distance estimators becomes class invariant unlike the several existing studies (Haseeb et al., 2018; Zhu & Fang, 2019).

Finally, we propose a novel CSEN architecture in the CL-CSEN model by introducing the ability to fine-tune the proxy mapping matrix during the training procedure. Therefore, the proposed CL-CSEN method is a complete, one-to-one support estimator network in which the denoiser matrix **B** is directly connected to the convolutional layers using fully connected dense layers. Thus, it provides a superior distance estimation performance and efficient single-stage inference. Overall, it is observed that CSEN and CL-CSEN architectures significantly outperform the competing methods used in this study. Finally, with their compact network models, we have shown that both CSEN and CL-CSEN are able to learn with a limited number of annotated data, e.g., with less than 13% annotated data used in the training to demonstrate this competence.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

KITTI 3D Object Detection dataset is available at http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d and we share the implementation at https://github.com/meteahishali/CSENDistance.

Acknowledgments

This work was supported by the NSF-Business Finland Center for Visual and Decision Informatics (CVDI) under the project Advanced Machine Learning for Industrial Applications (AMaLIA).

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: A system for large-scale machine learning. In *USENIX symp. operating syst. des. implementation* (pp. 265–283).
Ahishali, M., Degerli, A., Yamac, M., Kiranyaz, S., Chowdhury, M. E., Hameed, K., et al. (2021). Advance warning methodologies for covid-19 using chest x-ray images. *IEEE Access*, 9, 41052–41065.
Borgerding, M., Schniter, P., & Rangan, S. (2017). AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16), 4293–4308.
Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1), http://dx.doi.org/10.1561/2200000016.
Candes, E. J. (2008). The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathematique*, 346(9–10), 589–592.
Candes, E. J., & Plan, Y. (2011). A probabilistic and RIPless theory of compressed sensing. *IEEE Transaction on Information Theory*, 57(11), 7235–7254.
Candes, E., & Romberg, J. (2005). l_1-magic: Recovery of sparse signals via convex programming. *Technical Report*, Caltech, URL https://statweb.stanford.edu/~candes/software/l1magic/downloads/l1magic.pdf.
Casser, V., Pirk, S., Mahjourian, R., & Angelova, A. (2019). Depth prediction without the sensors: Leveraging structure for unsupervised learning from monocular videos. In *AAAI conf. artif. intell..*
Chang, J.-R., & Chen, Y.-S. (2018). Pyramid stereo matching network. In *Proc. IEEE conf. comput. vision pattern recognit.* (pp. 5410–5418).
Chen, S. S., Donoho, D. L., & Saunders, M. A. (2001). Atomic decomposition by basis pursuit. *Society for Industrial and Applied Mathematics*, 43(1), 129–159.
Deng, W., Hu, J., & Guo, J. (2018). Face recognition via collaborative representation: Its discriminant nature and superposed representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10), 2513–2521. http://dx.doi.org/10.1109/TPAMI.2017.2757923.
Donoho, D. L., & Elad, M. (2003). Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5), 2197–2202.
Figueiredo, M. A., Nowak, R. D., & Wright, S. J. (2007). Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4), 586–597.
Fletcher, A. K., Rangan, S., & Goyal, V. K. (2009). Necessary and sufficient conditions for sparsity pattern recovery. *IEEE Transaction on Information Theory*, 55(12), 5758–5772.
Geiger, A., Lenz, P., & Urtasun, R. (2012). Are we ready for autonomous driving? The KITTI vision benchmark suite. In *Proc. IEEE conf. comput. vision pattern recognit..*
Gökçe, F., Üçoluk, G., Şahin, E., & Kalkan, S. (2015). Vision-based detection and distance estimation of micro unmanned aerial vehicles. *Sensors*, 15(9), 23805–23846.
Guha, T., & Ward, R. K. (2011). Learning sparse representations for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(8), 1576–1588.
Haseeb, M. A., Guan, J., Ristić-Durrant, D., & Gräser, A. (2018). DisNet: a novel method for distance estimation from monocular camera. In *Workshop plann., percept. navig. intell. veh..*
He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proc. IEEE conf. comput. vision and pattern recognit.* (pp. 770–778).
Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proc. IEEE conf. comput. vision and pattern recognit.* (pp. 4700–4708).
Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv:1412.6980.
Koh, K., Kim, S.-J., & Boyd, S. (2007). An interior-point method for large-scale l_1-regularized logistic regression. *Journal of Machine Learning Research*, 8, 1519–1555.
Li, W., & Du, Q. (2016). A survey on representation-based classification and detection in hyperspectral remote sensing imagery. *Pattern Recognition Letters*, 83, 115–123.
Mahjourian, R., Wicke, M., & Angelova, A. (2018). Unsupervised learning of depth and ego-motion from monocular video using 3d geometric constraints. In *Proc. IEEE conf. comput. vision pattern recognit.* (pp. 5667–5675).
Malioutov, D. M., Cetin, M., & Willsky, A. S. (2005). Homotopy continuation for sparse signal representation. 5, In *Proc. IEEE int. conf. acoust., speech, and signal process.* (pp. 733–736).
Rad, K. R. (2011). Nearly sharp sufficient conditions on exact sparsity pattern recovery. *IEEE Transaction on Information Theory*, 57(7), 4672–4679.
Reeves, G., & Gastpar, M. (2008). Sampling bounds for sparse support recovery in the presence of noise. In *IEEE int. symp. inf. theory* (pp. 2187–2191).
Reeves, G., & Gastpar, M. (2012). The sampling rate-distortion tradeoff for sparsity pattern recovery in compressed sensing. *IEEE Transaction on Information Theory*, 58(5), 3065–3092.
Reeves, G., & Gastpar, M. C. (2013). Approximate sparsity pattern recovery: Information-theoretic lower bounds. *IEEE Transaction on Information Theory*, 59(6), 3451–3465.
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252.

Scarlett, J., & Cevher, V. (2016). Limits on support recovery with probabilistic models: An information-theoretic framework. *IEEE Transaction on Information Theory*, *63*(1), 593–620.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556.

Wainwright, M. (2007). Information-theoretic bounds on sparsity recovery in the high-dimensional and noisy setting. In *IEEE int. symp. inf. theory* (pp. 961–965).

Wang, W., Wainwright, M. J., & Ramchandran, K. (2008). Information-theoretic limits on sparse support recovery: Dense versus sparse measurements. In *IEEE int. symp. inf. theory* (pp. 2197–2201).

Wang, T.-H., Wang, F.-E., Lin, J.-T., Tsai, Y.-H., Chiu, W.-C., & Sun, M. (2019). Plug-and-play: Improve depth prediction via sparse data propagation. In *IEEE int. conf. robot. automat.* (pp. 5880–5886).

Williams, C., & Seeger, M. (2001). Using the nyström method to speed up kernel machines. In *Proc. 14th annu. conf. neural inf. process. syst.* (pp. 682–688).

Wright, J., Ma, Y., Mairal, J., Sapiro, G., Huang, T. S., & Yan, S. (2010). Sparse representation for computer vision and pattern recognition. *Proceedings of the IEEE*, *98*(6), 1031–1044.

Wright, J., Yang, A. Y., Ganesh, A., Sastry, S. S., & Ma, Y. (2008). Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(2), 210–227.

Xu, J., An, W., Zhang, L., & Zhang, D. (2019). Sparse, collaborative, or nonnegative representation: which helps pattern classification? *Pattern Recognition*, *88*, 679–688.

Yamaç, M., Ahishali, M., Degerli, A., Kiranyaz, S., Chowdhury, M. E., & Gabbouj, M. (2021). Convolutional sparse support estimator-based COVID-19 recognition from X-Ray images. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(5), 1810–1820.

Yamac, M., Ahishali, M., Kiranyaz, S., & Gabbouj, M. (2020). Convolutional sparse support estimator network (CSEN) from energy efficient support estimation to learning-aided compressive sensing. arXiv:2003.00768.

Yang, T., Li, Y.-F., Mahdavi, M., Jin, R., & Zhou, Z.-H. (2012). Nyström method vs random fourier features: A theoretical and empirical comparison. *Advances in Neural Information Processing Systems*, *25*, 476–484.

Yang, A. Y., Zhou, Z., Balasubramanian, A. G., Sastry, S. S., & Ma, Y. (2013). Fast l_1-minimization algorithms for robust face recognition. *IEEE Transactions on Image Processing*, *22*(8), 3234–3246.

Zhang, L., Yang, M., & Feng, X. (2011). Sparse representation or collaborative representation: Which helps face recognition? In *Proc. IEEE int. conf. comput. vision* (pp. 471–478).

Zhu, J., & Fang, Y. (2019). Learning object-specific distance from a monocular image. In *Proc. IEEE int. conf. comput. vision* (pp. 3839–3848).